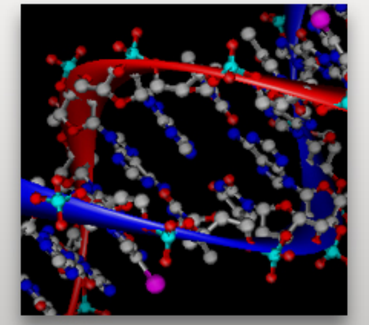
Installing and Running Secure Solutions in Mac OS X Tiger

Session 501

These are confidential sessions—please refrain from filming or taking pictures.



OS Foundations

Installing and Running Secure Solutions in Mac OS X Tiger

Ron Dumont, Manager, Product Security
Richard Murphy, Manager, Data Security
Aaron Sigel, Red Team Leader, Product Security
Simon Cooper, Senior Engineer, Product Security

http://developer.apple.com/wwdc2005/

What You'll Learn

- Security threats and trends
- Environment—think globally, act locally
- Data handling for security and privacy
- Common coding pitfalls and how to avoid them
- System services for security

Security Threats and Trends

Ron Dumont
Manager, Product Security

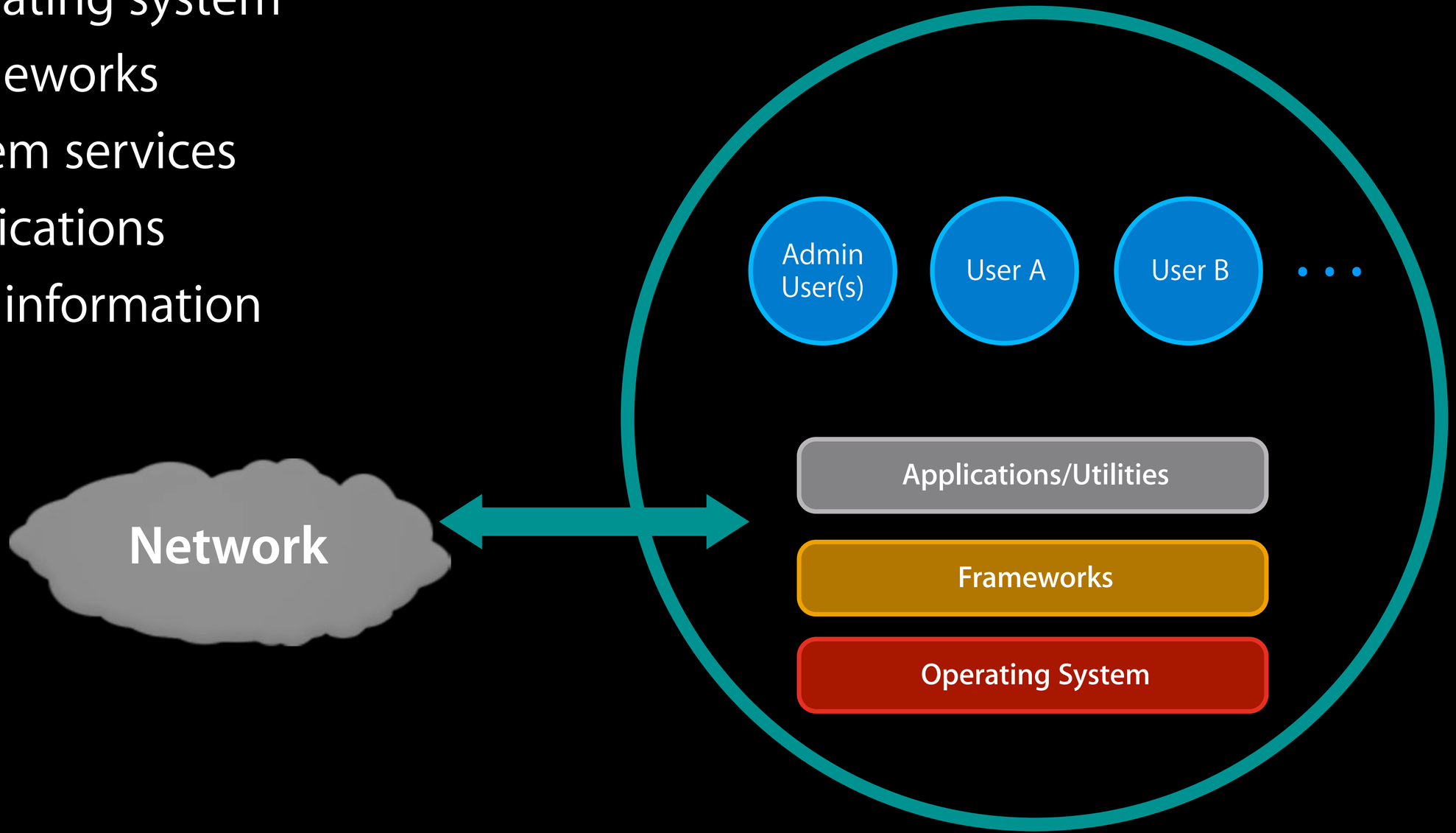
http://developer.apple.com/wwdc2005/

Introduction

- Attack vectors—the paths to compromising systems
- Security trends
- Not specific to any particular operating system

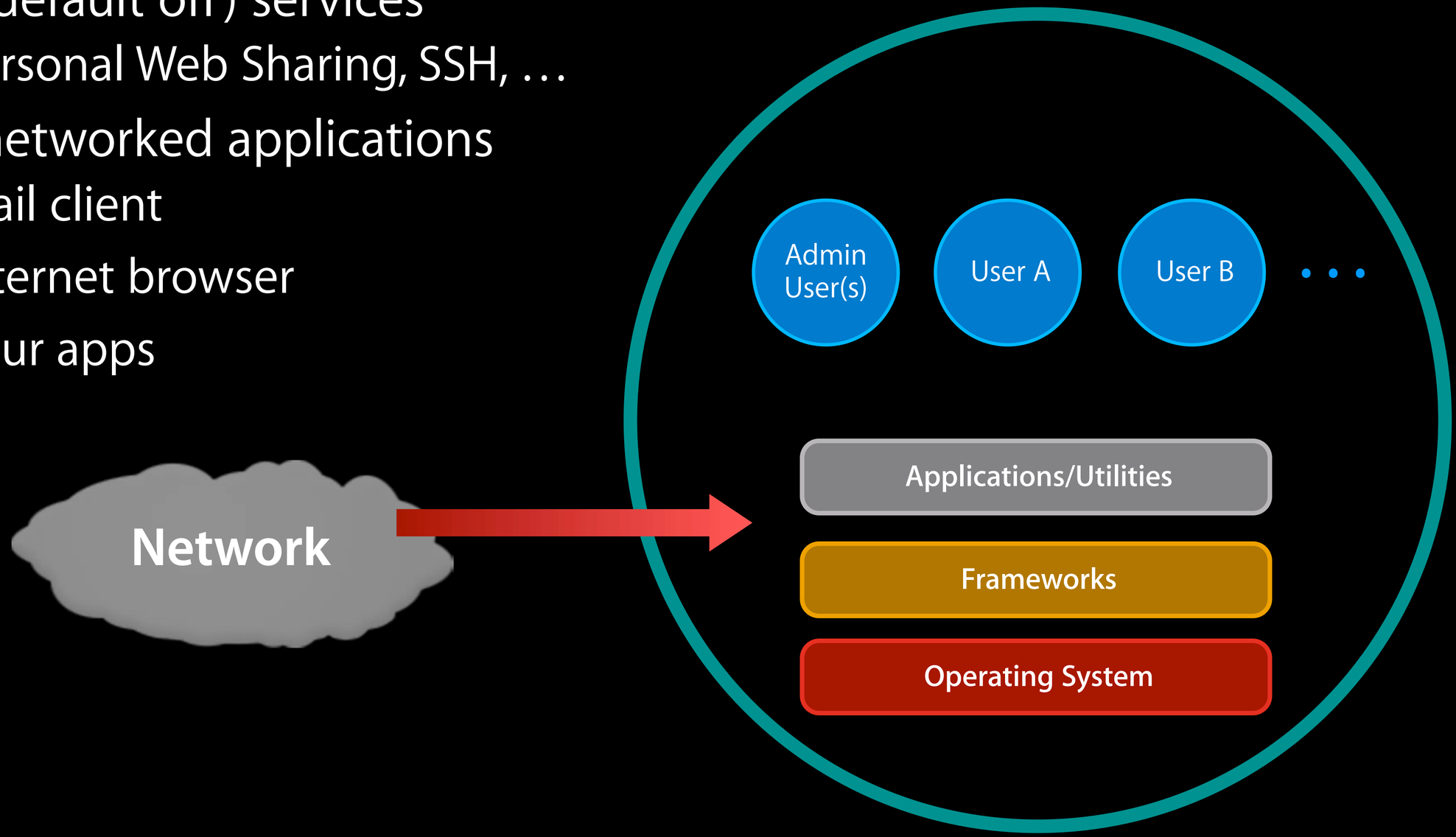
Attacking What?

- Operating system
- Frameworks
- System services
- Applications
- User information



Remote Attack

- Via (default off) services
  - Personal Web Sharing, SSH, …
- Via networked applications
  - Mail client
  - Internet browser
  - Your apps

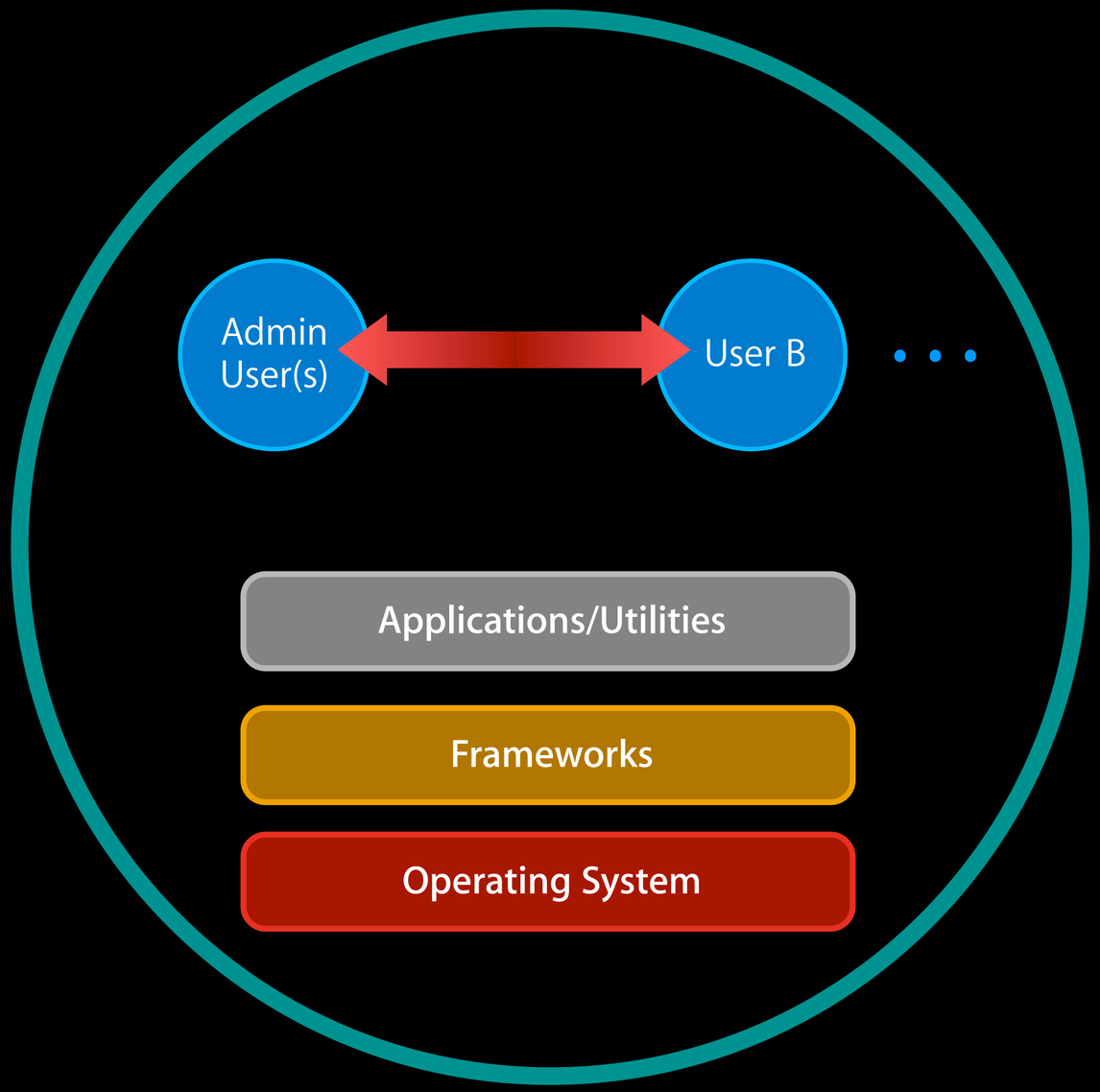


Remote Control?

- Open and listen on a port
- Update software via the network
- Process input from the network
- Register a URL handler
- Reveal information in response to network queries

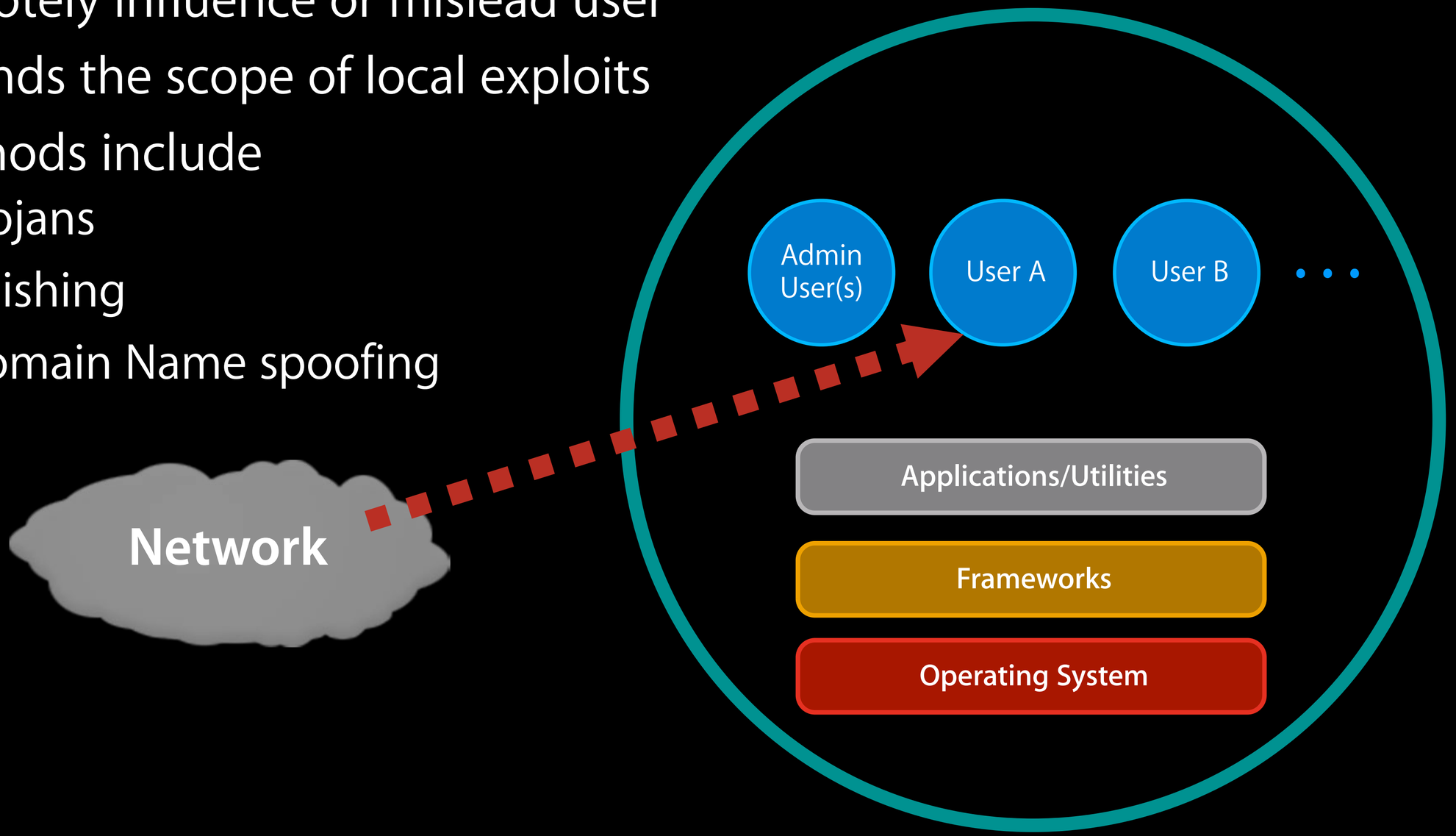
Local Attack

- Account on the system
- Physical access to the computer
- Exploit vulnerabilities
  - Privilege escalation
  - Weak passwords
  - Insecure user data



Social Engineering Attack

- Remotely influence or mislead user
- Extends the scope of local exploits
- Methods include
  - Trojans
  - Phishing
  - Domain Name spoofing



Points of Vulnerability

- Running with elevated privileges
- Be wary when allowing code plugins
- Storing valuable user information

Trends

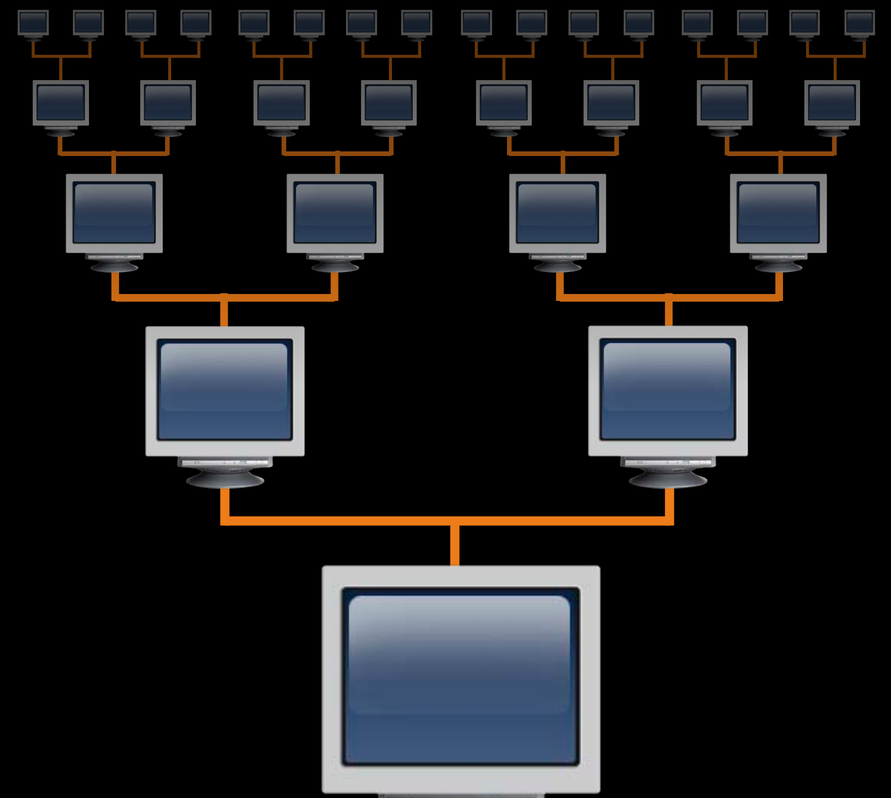
Increasingly Networked

- “Always on” internet connection
- Higher bandwidth
- Wireless at home, internet cafes, airports…
- Many new features rely on network connections

Most networks are insecure

Trends: Botnets

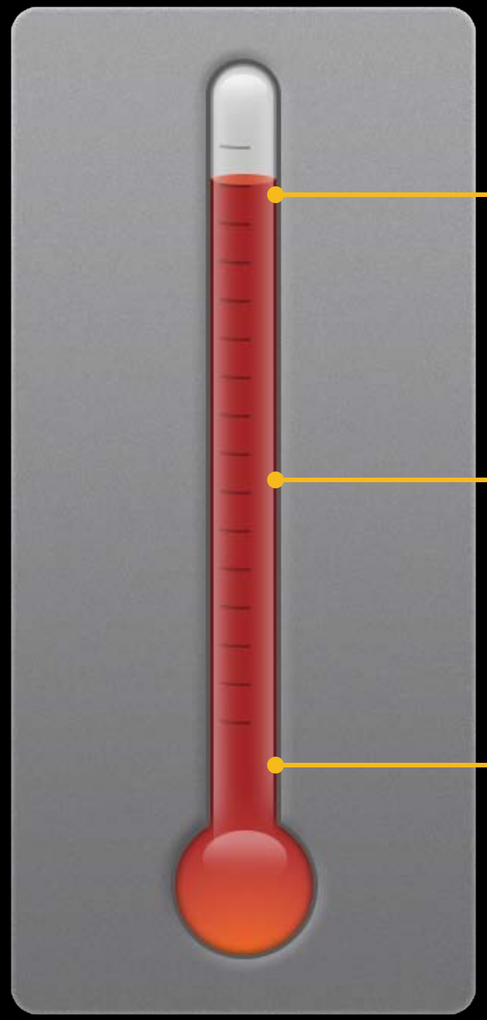
- It's no longer the sole hacker
- Botnet tools freely obtainable
- Exploit known vulnerabilities, open ports, weak passwords
- Botnets are self-replicating
  - ≈ 1000—Script Kiddies
  - ≈ 50,000—Organized crime
- Sophisticated Rootkits to secure hacked systems
  - Disables automatic Software Update
  - Installs a keystroke recorder and network sniffer
  - Adds backdoor accounts and enables network services
  - …and much more



March 30, 2005 4:03 PM
Chinese Authorities Arrest Super Hacker

CHINESE AUTHORITIES said they've arrested a man they accuse of hacking into **100,000** computers. Coordinated attacks were manipulated by the hacker after he gained access to the PCs through the use of a virus. Up to "40,000 of the affected machines may be abroad," the Ministry said.

Attack Trends



Theft
identity, money, application data…

Distributed Denial of Service
for hire, extortion or eliminating competition

OS-level attacks

Exploits

- Less technical ability required to attack
  - Only one hacker needs to write an exploit
  - Proof of concept tools transfer skill
- Exploits for publicly disclosed issues appear faster
- Decreasing notification time before exploits are released
  - 0-Day exploits
- Most problems come from old, unpatched systems

Summary

- Remote, local, and social engineering attacks
- Networking
  - Always-on, faster, wireless, features rely on it
- Botnets
  - Powerful tool for attacks and theft
- Attacks focusing on information theft
- Exploits and attack tools are getting more sophisticated
  - One person hacks, everyone gets the sploit

Mac OS X Is Multi-User

Richard Murphy
Manager, Data Security

http://developer.apple.com/wwdc2005/

Home Directories

- Each user has a home directory in (by default) /Users
  - Locations to store persistent and temporary personal data
    - Documents
    - ~Library
      - Preferences
      - Logs
      - Caches
      - By default has more restricted access
  - Public
    - Shared Data Access

Home Directories

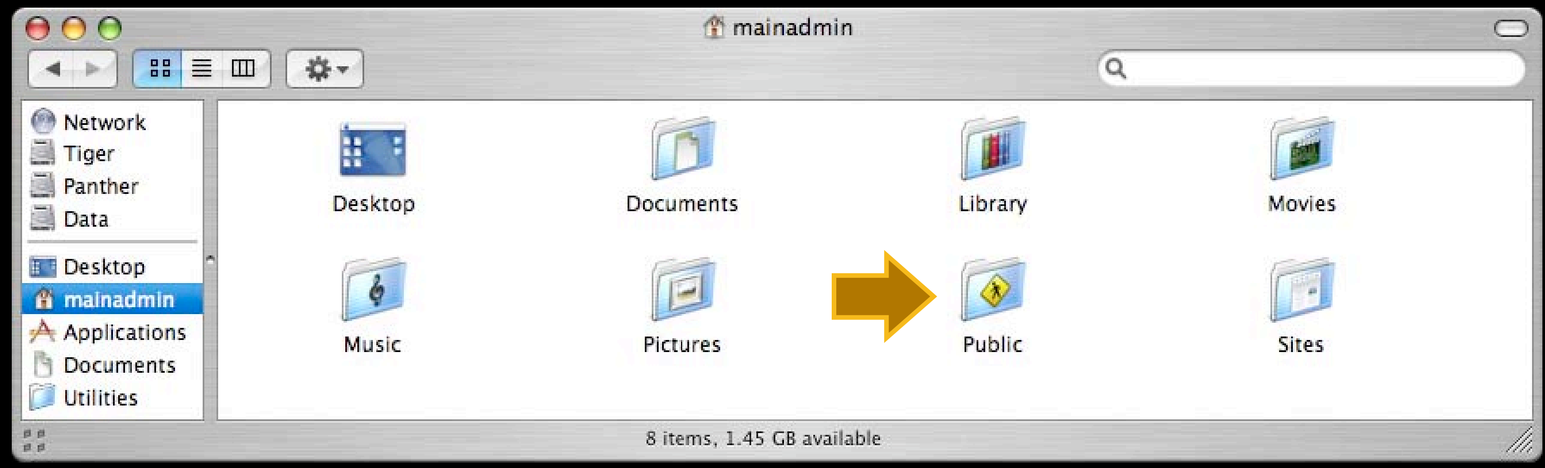
- Each user has a home directory in (by default) /Users
  - Locations to store persistent and temporary personal data
    - **Documents**
    - ~Library
      - Preferences
      - Logs
      - Caches
      - By default has more restricted access
  - Public
    - Shared Data Access

Home Directories

- Each user has a home directory in (by default) /Users
  - Locations to store persistent and temporary personal data
    - Documents
    - ~Library
      - Preferences
      - Logs
      - Caches
      - By default has more restricted access
    - Public
      - Shared Data Access

Home Directories

- Each user has a home directory in (by default) /Users
  - Locations to store persistent and temporary personal data
    - Documents
    - ~Library
      - Preferences
      - Logs
      - Caches
      - By default has more restricted access
  - **Public**
    - Shared Data Access



Home Directories

- Each user has a home directory in (by default) /Users
  - Locations to store persistent and temporary personal data
    - Documents
    - ~Library
      - Preferences
      - Logs
      - Caches
      - By default has more restricted access
    - Public
      - Shared Data Access
    - **Music and Movies**

Home Directory Types

- “Normal”
  - The default directory type
- FileVault
  - Not mounted when users are logged out
  - Storing files in FileVault adds protection
  - Unreadable by other users by default
- Network Home Directories
  - Can be accessed by multiple systems simultaneously
  - Those systems can be very different
    - Preferences might be affected by system type

Play Well With the User's Home Directory!

- Don't change permissions of files or directories unrelated to your application
  - We've seen applications "loosen" whole directory trees to make operations easier
- Use the provided directories where you can
  - Default document storage to the Documents folder, Music to the Music folder, etc.
    - Easier to backup
    - Easier to locate info
    - Makes the perimeter of things that need to be secured "tighter"
- The user controls all data in their home directory
  - Never assume a user hasn't manually edited a preference or configuration file in their home directory

Fast User Switching

- Don’t assume the user of your application is the only person using the system or that application
- Applications can be running while another user is active
- Be careful using shared memory segments or files not to share across users!
- If your application creates resources or controls them—decide how a user gains control of them

System Resources

- Some applications are system level resources
- In this case system areas should be used to caches, logging and temp space
- Long term processes need to daemonize and move to the root bootstrap

StartupItems

- /Library/StartupItems is for System wide services
  - Must be admin to install
  - Don’t put “faced” applications there
    - We’ve had numerous security reports on this sort of activity
    - We’ve tightened permissions on the directory

```
gandalf:/Library/StartupItems/badstartupperms murf$ ls -la
total 16
drwxrwxrwx   5 root  admin  170 May 25 11:05 .
drwxr-xr-x   4 root  wheel  136 May 25 10:51 ..
-rw-r--r--   1 root  wheel    0 May 25 11:05 .disabled
-rwxrwxrwx   1 root  admin  584 May 25 10:56 StartupParameters.plist
-rwxrwxrwx   1 murf  admin  405 May 25 10:55 badstartuperms
gandalf:/Library/StartupItems/badstartupperms murf$ _
```

- Login StartupItems is per-user
  - Can be installed by “normal users”
  - May be applications with “faces”

Admin/“Normal” Users

- Not all users are “Admins”
  - The “default user” setup is admin
- Make sure your application works when logged in to a non-admin account
- Installs may require admin, normal operations probably shouldn’t
- Security Policy: No escalation to root privilege without authorization

Authorization

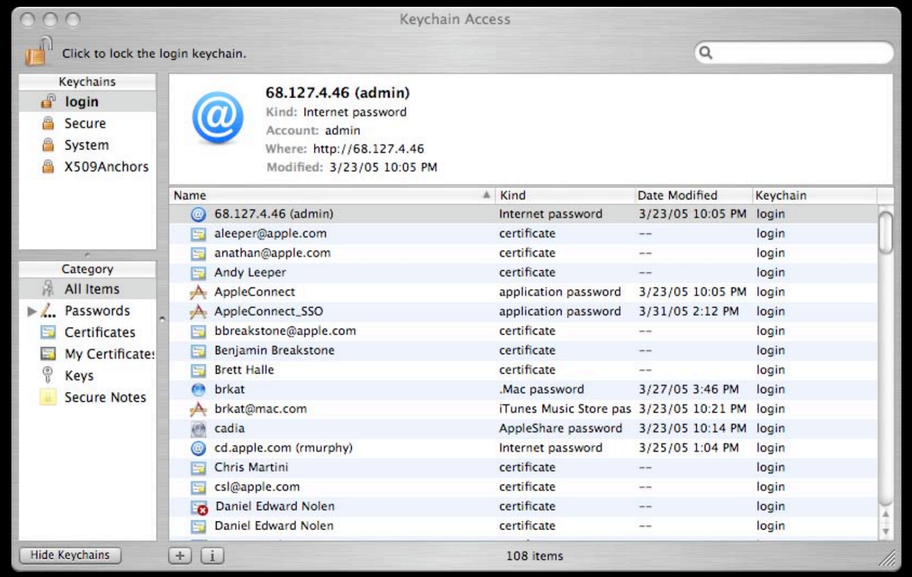
- Use the existing SecAuth interfaces to authenticate a user
- Don’t assume a password is used to authenticate
  - Tiger has improved smart card support
  - We’re investigating other forms of authentication
  - Use SecAuth to “future-proof” your application!

Keychains

- Normal Keychains are per-user
- Users may have more than one
  - Don’t assume the login keychain is the only one to search
  - Setup ACLs for your items for convenient access control
  - Some users are more “security conscious” than others
  - Even their login keychains may be locked
  - They may never set an ACL for automatic access

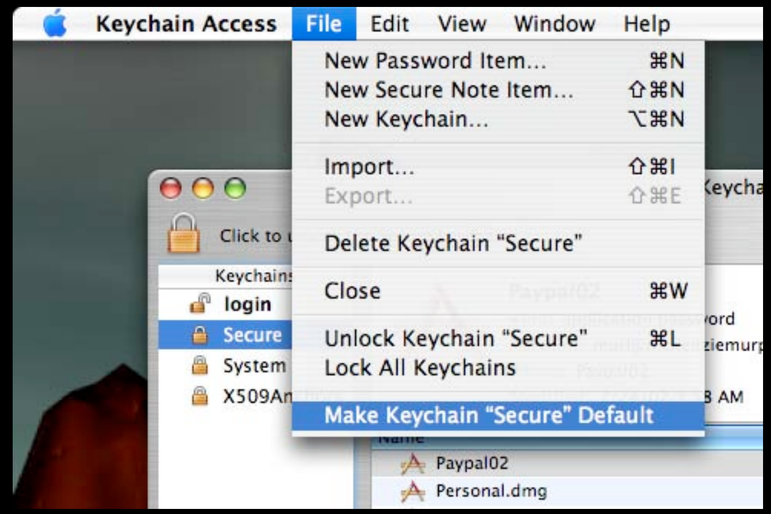
Keychain Special Types

- **Login Keychain**
  Automatically unlocked at login
- **Default Keychain**
  Default place new keychain items are saved
- **X509Anchors**
  System-wide Root Certificate repository
- **SystemKeychain**
  System-wide password storage available to root session



Keychain Special Types

- **Login Keychain**
  Automatically unlocked at login
- **Default Keychain**
  Default place new keychain items are saved
- **X509Anchors**
  System-wide Root Certificate repository
- **SystemKeychain**
  System-wide password storage available to root session



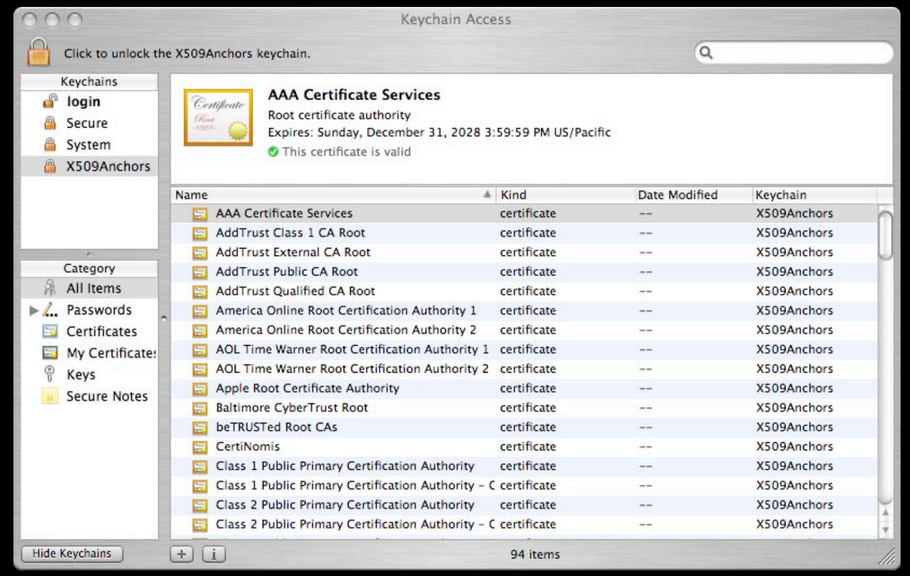
Keychain Special Types

- **Login Keychain**
  Automatically unlocked at login

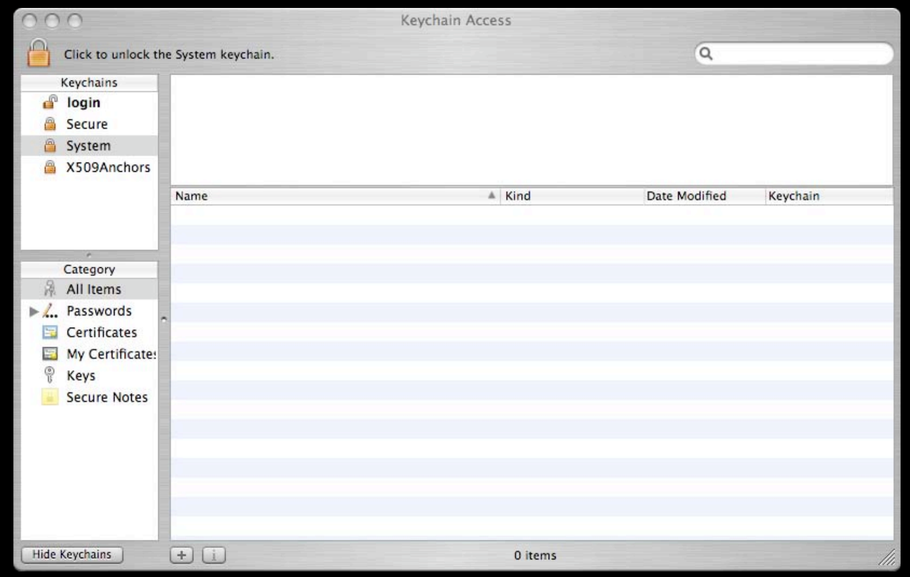
- **Default Keychain**
  Default place new keychain items are saved

- **X509Anchors**
  System-wide Root Certificate repository

- **SystemKeychain**
  System-wide password storage available to root session



Keychain Special Types



- **Login Keychain**
  Automatically unlocked at login
- **Default Keychain**
  Default place new keychain items are saved
- **X509Anchors**
  System-wide Root Certificate repository
- **SystemKeychain**
  System-wide password storage available to root session

User Keychains

- Cannot have their passwords reset by anyone that doesn't know the original password. Period. End of story.
- securityd acts as the "arbiter" of keychain access control—keychains are not accessible in normal process address space—keychain items are passed from the securityd to the application
  - Passwords are retreived by the securityd
  - Access to the password is granted after an ACL check
  - All private key operations are performed inside the securityd
    - Private Keys are not normally handed to applications

User Keychains

- Are excellent places to keep passwords that need to be handed out in a controlled manner
- Are cryptographically strong for the “private” portion of an item

Data Handling

Aaron Sigel
Red Team Leader, Product Security

http://developer.apple.com/wwdc2005/

Why Should You Care?

Your reputation is on the line!

What Are We Talking About?

Privacy

- Keeping sensitive information safe from disclosure

Protection

- Keeping the data safe from modification
- Keeping the data safe from destruction

Privacy

What should be private?

- Financial data
- Intellectual Property
- Passwords
- Information on your browsing habits

Privacy

Common pitfalls

- Insecure BSD file permissions
- Trusting BSD file permissions means trusting admin
- Sensitive information in error messages and log files
- Encryption
  - Not using encryption
  - Not using appropriate encryption techniques (snake oil)
- False expectations

Protection

What should be protected?

- Securing data against modification
  - Reports, messages, financial data, etc.
  - Personal information
  - Application data
- System configuration
  - Privilege escalation
  - Silent security degradation
- Securing data against destruction
  - What if your application's user data was destroyed?

Protection

Common pitfalls

- BSD file permissions
  - BSD file permissions help prevent tampering
  - Admin users can override BSD file permissions
  - Only cryptographic signing can assure you data is untampered
- Insufficient integrity checking and input validation
- Not error checking

Directory Layout

Application

- Applications usually live in /Applications or ~/Applications
- Applications should not write in their bundle
- Root owner on applications combats malware
- Updating should require authorization

/Applications/MyApplication.app
Where the application itself lives

- `r-xr-x-r-x`
- Owned by root:wheel

Directory Layout

Startup Items

- Startup items must have the correct permissions to work
- Security Fixer (in Tiger) will identify incorrect permissions
- This is where Opener attacked

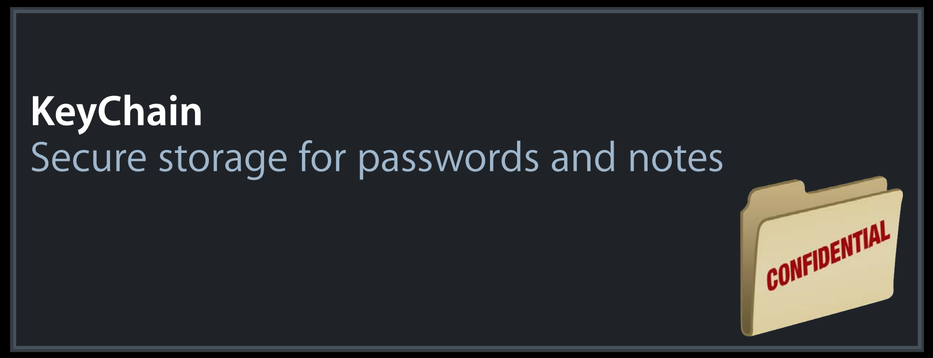
/Library/StartupItems or
/Library/LaunchDaemons
Where startup items live

- `r-xr-x-r-x`
- Owned by root:wheel

Directory Layout

Securely storing credentials

- Keychains are portable
- No cleartext passwords on disk
- Privacy from system administrators
- User is alerted when new applications attempt to access them
- Not appropriate for large files…



Directory Layout

Securely storing data

- BSD Permissions must also be used to block destruction
- Disk Images are portable
- Privacy from system administrators

Encrypted Disk Images
Secure storage for data files

Directory Layout

User preferences, plug-ins, caches, and logs

- Take advantage of the home directory
- ~Library has restrictive permissions by default
- Keep all user-specific resources in the home directory
- Sensitive data should still be encrypted

~/Library/Preferences
User specific preferences

- `rwx------`
- Owned by User1

~/Library/Caches
Temporary content caching

- `rwx------`
- Owned by User1

~/Library/Application Support
App plug-ins, support files, and resources

- `rwx------`
- Owned by User1

~/Library/Logs
User log files

- `rwx------`
- Owned by User1

Directory Layout

System wide preferences and resources

- Store only system wide preferences, plug-ins, and resources in /Library
- No expectation of privacy on these files

/Library/Preferences
User specific preferences

- `rw-r--r--`
- Owned by root

/Library/Application Support
App plug-ins, support files, and resources

- `rwxr-xr-x`
- Owned by root

Privacy and Protection Solutions

Privacy	Protection	Local Data	Network Data	Technology
●	●		●	Secure Transport
●		●		Encrypted Disk Images
●		●		FileVault
●	●	●		Keychain
	○ Not Cryptographic!	●		BSD Permissions
●	●		●	IPSEC
●	●	●	●	CDSA

Additional information
http://developer.apple.com

Questions
cdsa-apple@group.apple.com

Secure Coding and Installation

Simon Cooper
Senior Engineer, Product Security

http://developer.apple.com/wwdc2005/

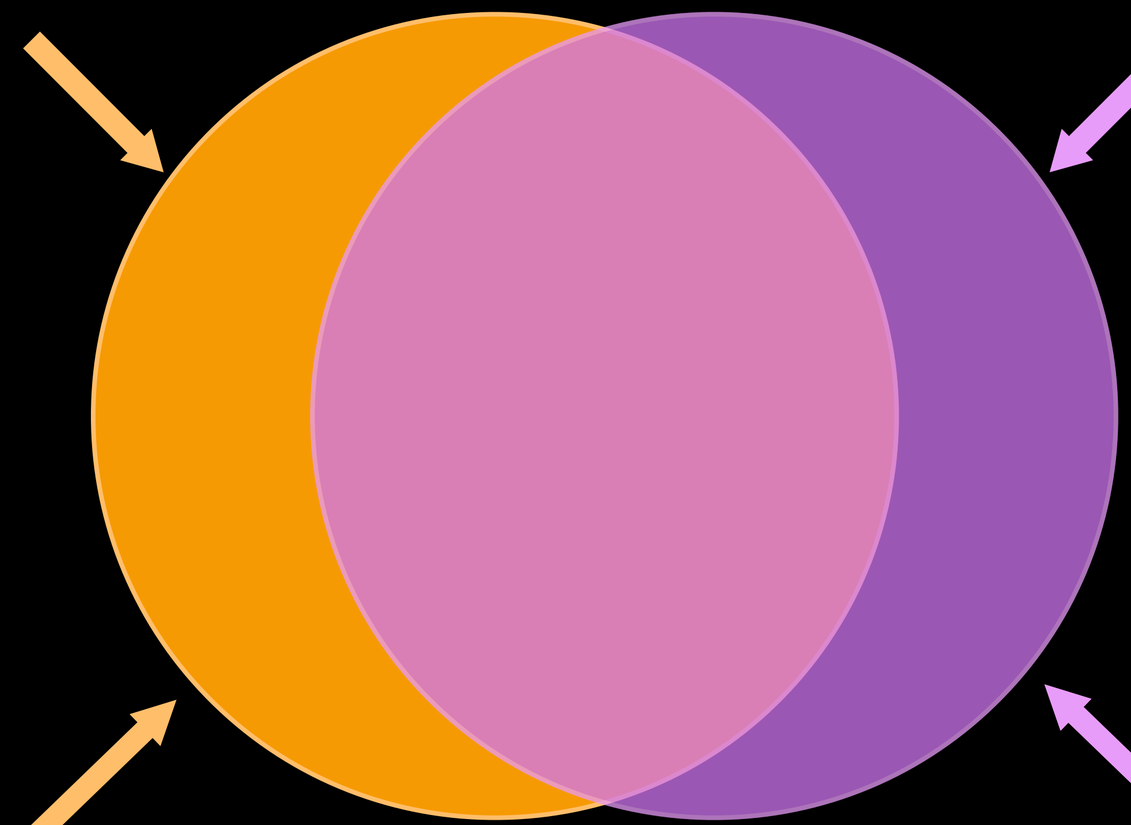
What Will be Covered?

- Where do security problems come from?
- Common coding and design pitfalls
  - Buffer/integer overflows
  - Temporary files (race conditions)
  - Feature combining/complexity
- Installation considerations

Where Do Security Problems Come From?

Missing or broken functionality found by QA when program does not behave as expected with good test data

Unexpected behavior and features found by giving program malformed data or if it is run in a non-usual way. Hacker territory



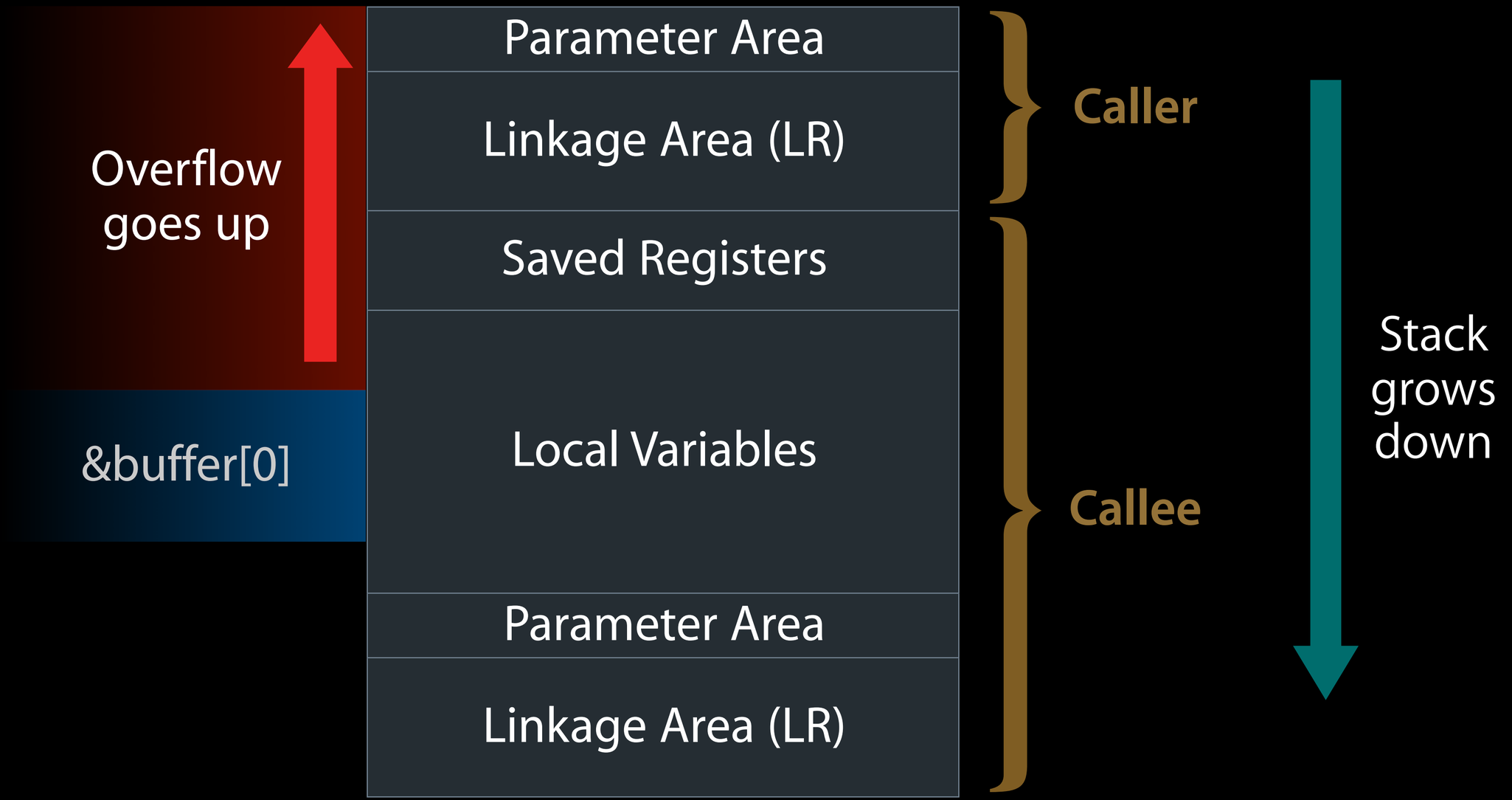
Specification

Implementation

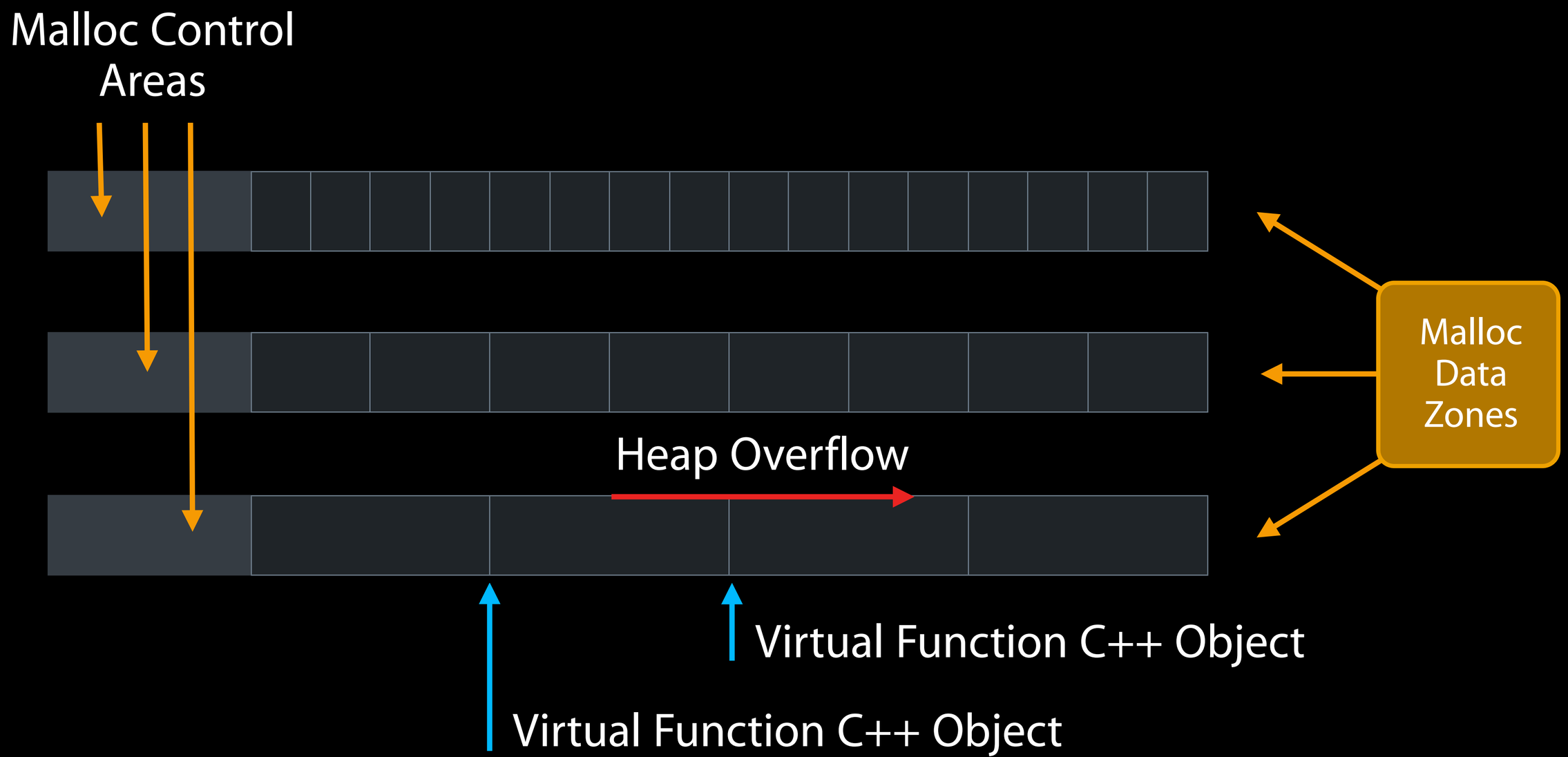
Buffer and Integer Overflows

- #1 root cause of security flaws
- Buffers always have a size and is >= 0
- When things go wrong…
  - Stack overflow
    - Execution path modified
    - Function arguments, local data modified
    - Usually trivial to exploit
  - Heap overflow
    - Program data modified
    - Execution path modified
    - Dangerous for C++ code
  - The result of malloc(0) is not 0

Mac OS X PPC Stack Frame



Heap Overflow



Buffer and Integer Overflows—Avoiding

- Languages we'll cover:
  - C, C++ and Objective C
  - Use of framework buffer handling routines
- For safer C buffer handling
  - C functions to use
  - C functions to avoid
  - C coding style to use
  - C coding style to avoid

String Functions to Avoid and Use

When you think/see these functions	Use these functions instead
strcat, strcpy	strlcat, strlcpy
strncat, strncpy	strlcat, strlcpy
sprintf, vsprintf	snprintf, vsnprintf
gets	fgets

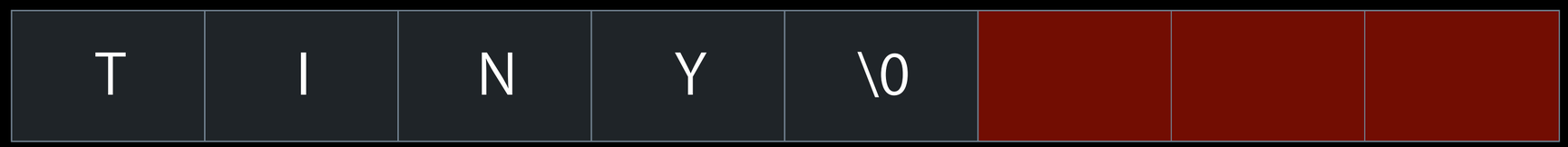
Always use unsigned variables for calculating sizes

str{cpy,cat} vs. strn{cpy,cat} vs. strl{cpy,cat}

```
char destination[5]; char *source = “TINY”;

strcpy(destination, source);
```

```
strncpy(destination, source, sizeof(destination));
```




```
strlcpy(destination, source, sizeof(destination));
```

str{cpy,cat} vs. strn{cpy,cat} vs. strl{cpy,cat}

```
char destination[5]; char *source = “LARGER”;

strcpy(destination, source);
```

```
strncpy(destination, source, sizeof(destination));
```

```
strlcpy(destination, source, sizeof(destination));
```

C Coding Styles to Avoid

Avoid

```
char buf[1024];
...
if (size <= 1023) {
   ...
}
```

OR

```
char buf[1024];
...
if (size < 1024) {
   ...
}
```

Use

```
char buf[BUF_SIZE];
...
if (size < BUF_SIZE) {
   ...
}
```

OR

```
char buf[1024];
...
if (size < sizeof(buf)) {
   ...
}
```

C Coding Styles to Avoid

Avoid

```
{
   char file[MAX_PATH];
...
   addsfx(file);
...
}

static *suffix = “.ext”;

char *addsfx(char *buf)
{
    return strcat(buf, suffix);
}
```

Use

```
{
   char file[MAX_PATH];
...
   addsfx(file, sizeof(file));
...
}

static *suffix = “.ext”;

char *addsfx(char *buf, unsigned int size)
{
    return strlcat(buf, suffix, size);
}
```

C++, Objective C and Frameworks

- For C++ STL string handling avoids buffer overflows
  - Problem: does not deal with encoding type—char, wchar
- For Objective C use NSString
  - Need care when converting to C strings
  - Passing to POSIX APIs may require conversion
    - e.g. [NSString fileSystemRepresentation]
- Foundation has CFString
  - Similar to NSString
  - Can be used in plain C code

Integer Overflow

- Arithmetic on sizes can be surprising!

```
int 2147483647 + 1 = - 2147483648
    0x7fffffff + 1 = 0x80000000

1024 + 2147483647 < 1024

n * m = 0 mod 2**32  (for sets of values of n and m)
```

- A problem if `n` and/or `m` come from user suppled data
- Checks

```
size_t bytes = n * m;

if (n != 0 && m != 0 && bytes / m == n) {
   ... /* allocate “bytes” space */
}
```

- Place limits on user supplied `n` and/or `m` to avoid overflows

Temporary Files

- Publicly writable places are not safe
- Symbolic links
  - Time of check, time of use—race conditions
- Running as root != running as user
- Securely creating files in 0777 places
- Securely creating directories in 0777 places
- Use: mkstemp() with at least 6 ‘X’ chars
  - This results in `56,800,235,584 (62 ** 6)` possible names

```
int *temporary = mkstemp(“tmp/myTmp.XXXXXX”)
```

Temporary Files

```
FILE *safe_open_wplus(char *fname) {
  struct stat lstat_info, fstat_info;
  FILE *fp;
  char *mode = “rb+”; /* Truncation is performed */
  int  fd;
  if (lstat(fname, &lstat_info) == -1) {
     if (errno != ENOENT) { return 0; }
     if ((fd = open (fname, O_CREAT|O_EXCL|O_RDWR, 0600)) == -1) { return 0; }
     mode = “wb”;
  } else {
     if ((fd = open (fname, O_RDWR)) == -1) { return 0; }
     if (fstat(fd, &fstat_info) == -1 ||
         lstat_info.st_mode != fstat_info.st_mode ||
         lstat_info.st_ino  != fstat_info.st_ino ||
         lstat_info.st_dev  != fstat_info.st_dev) {
            close (fd);
            return 0;
     }
     ftruncate (fd, 0);
  }
  fp = fdopen(fd, mode);
  if (!fp) { close (fd); unlink (fname); return 0; }
  return fp;
}
```

Feature Combining

- Feature 1 is awesome
- Feature 2 is wonderful
- Feature 1 + Feature 2 = awesomely wonderful security hole

Complexity

- Has that code path ever been tested?
- Backwards compatibility, does it downgrade security?
- A larger source base contains more bugs
- A larger source base is harder to maintain

Installation Considerations

- Use secure permissions when installing
- Don’t allow modification of your application bundle
- Don’t modify the permissions of system components
- Use pre- and post-install scripts sparingly
- Pre- and post-install scripts need to be written securely

Summary

- Attack trends
  - Increasingly sophisticated, automated, focused on information theft
  - Only 1 person needed to create an exploit—explosively replicated
- Multi-User considerations
  - Use the correct locations to store persistent and temporary data
  - Use keychains for secure storage of passwords and secrets
- Mac OS X system services for information privacy and protection
- Secure coding
  - Buffer overflows are the #1 flaw
  - Coding safely provides enhanced security and improved software quality
  - Installation is part of the security process

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Who to Contact

Craig Keithley
Security Technology Evangelist
keithley@apple.com

Q&A

Garth Cummings
Developer Technical Support

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference